1. **Unipolar NRZ**

```
clear all;close all;clc;

bit_d = 1e-3;% bit duration in sec
fc = 10000;% frequency of the carrier in Hz
fs = 10*fc;% Sampling rate
%-----------------------------------------------------------
n = 0:1:(bit_d*fs)-1;% Discrete time
carr1 = cos(2*pi*(fc/fs)*n);% Carrier wave for one bit duration
digi_data = [1 0 0 1 1 0 1 0];% digital data to be transmitted
carr = repmat(carr1,size(digi_data));% Carrier for entire
t = 0:1/fs:(length(carr)-1)/fs;% time for entire duration o
%------------------------------------------------------------
enc_digi = [];
for i = 1:length(digi_data)
 if(digi_data(i)==1)
 enc_digi =[enc_digi ones(1,length(n))];
 else
 enc_digi =[enc_digi zeros(1,length(n))];
 end
end
%------------------------------------------------------------
figure(1),subplot(211),plot(t,enc_digi,'LineWidth',2);
xlabel('time in sec');
ylabel('Amplitude');
title('Digital Data');
enc_data = [];
for i = 1:length(digi_data)
 if(digi_data(i)==1)
 enc_data =[enc_data 1*ones(1,length(n))];
 else
 enc_data =[enc_data 1*zeros(1,length(n))];
 end
end
%-----------------------------------------------------------
figure(1),subplot(212),plot(t,enc_data,'LineWidth',2);
xlabel('time in sec');
ylabel('Amplitude');
title('Unipolar NRZ');
```

2. Polar NRZ signal

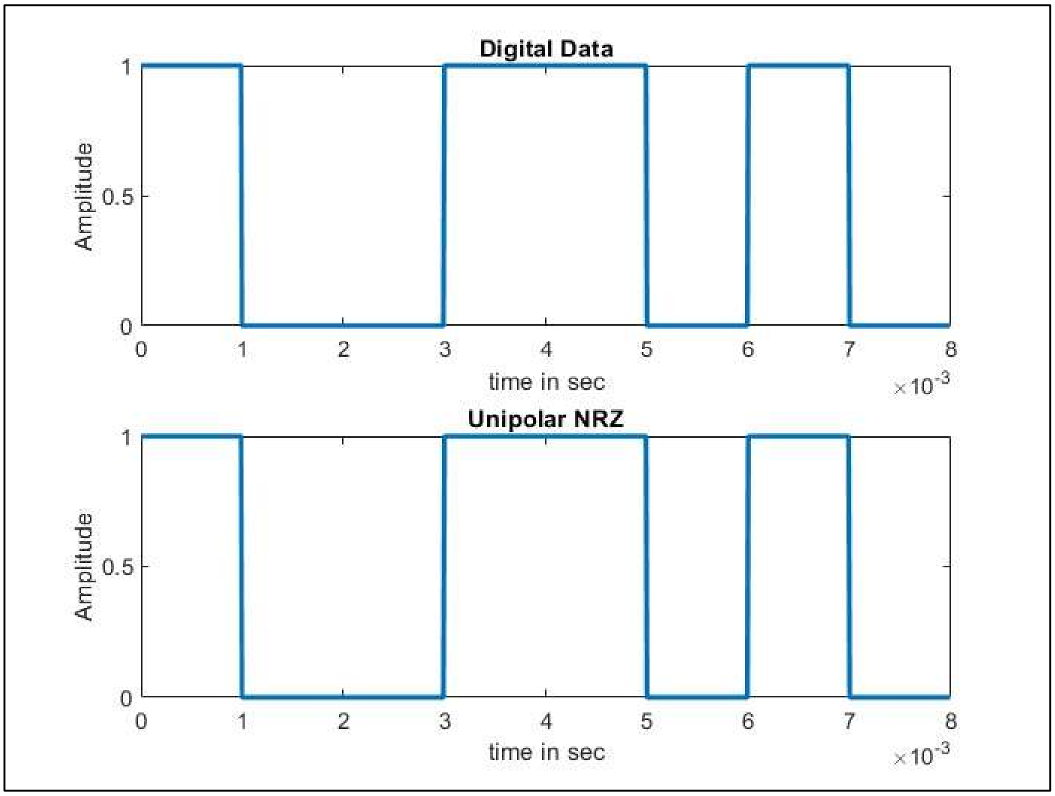
```
clear all;close all;clc;

bit_d = 1e-3;% bit duration in sec
fc = 10000;% frequency of the carrier in Hz
fs = 10*fc;% Sampling rate
%-----------------------------------------------------------
n = 0:1:(bit_d*fs)-1;% Discrete time
carr1 = cos(2*pi*(fc/fs)*n);% Carrier wave for one bit duration
digi_data = [1 0 0 1 1 0 1 0];% digital data to be transmitted
carr = repmat(carr1,size(digi_data));% Carrier for entire
t = 0:1/fs:(length(carr)-1)/fs;% time for entire duration o
%-----------------------------------------------------------
enc_digi = [];
for i = 1:length(digi_data)
 if(digi_data(i)==1)
 enc_digi =[enc_digi ones(1,length(n))];
 else
 enc_digi =[enc_digi zeros(1,length(n))];
 end
end
%----------------------------------------------------------
figure(1),subplot(211),plot(t,enc_digi,'LineWidth',2);
xlabel('time in sec');
ylabel('Amplitude');
title('Digital Data');
enc_data = [];
for i = 1:length(digi_data)
 if(digi_data(i)==1)
 enc_data =[enc_data 1*ones(1,length(n))];
 else
 enc_data =[enc_data -1*ones(1,length(n))];
 end
end
%-----------------------------------------------------------
figure(1),subplot(212),plot(t,enc_data,'LineWidth',2);
xlabel('time in sec');
ylabel('Amplitude');
title('Polar NRZ');
```

3. Bipolar NRZ

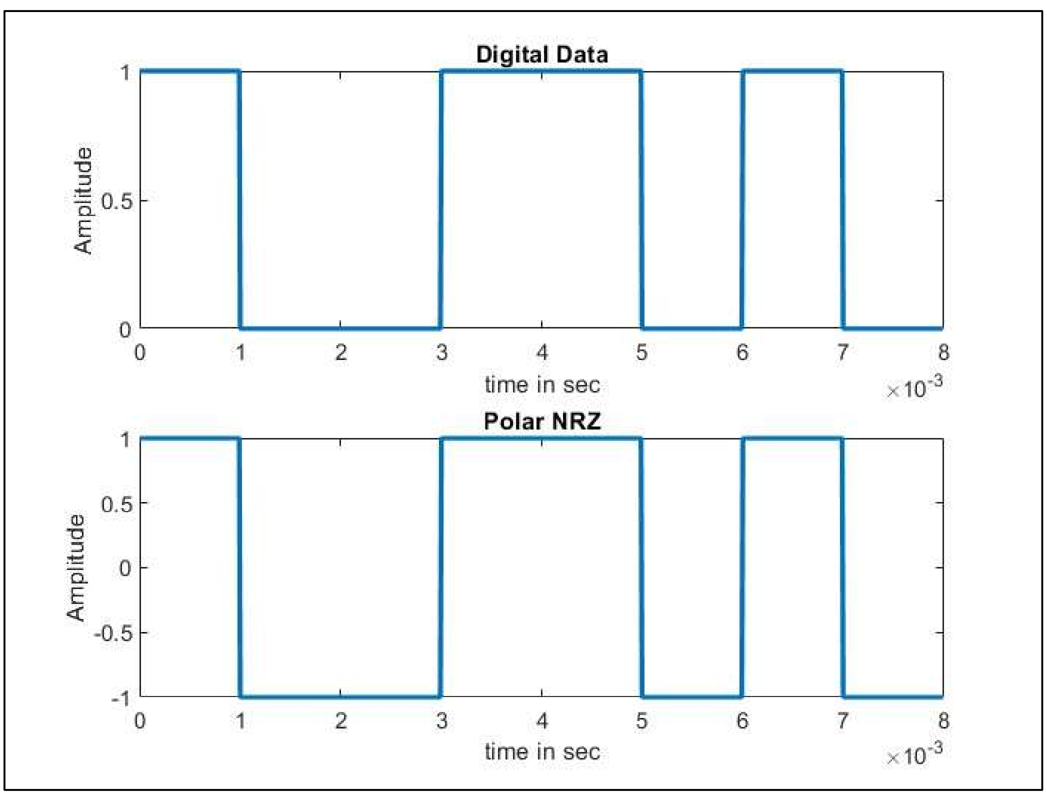
```
clear all;close all;clc;
m=2;

bit_d = 1e-3;% bit duration in sec
fc = 10000;% frequency of the carrier in Hz
fs = 10*fc;% Sampling rate
%-----------------------------------------------------------
n = 0:1:(bit_d*fs)-1;% Discrete time
carr1 = cos(2*pi*(fc/fs)*n);% Carrier wave for one bit duration
digi_data = [1 0 0 1 1 0 1 0];% digital data to be transmitted
carr = repmat(carr1,size(digi_data));% Carrier for entire
t = 0:1/fs:(length(carr)-1)/fs;% time for entire duration o
%----------------------
enc_digi = [];
for i = 1:length(digi_data)
 if(digi_data(i)==1)
 enc_digi =[enc_digi ones(1,length(n))];
 else
 enc_digi =[enc_digi zeros(1,length(n))];
 end
end
%-----------------------------------------------------
figure(1),subplot(211),plot(t,enc_digi,'LineWidth',2);
xlabel('time in sec');
ylabel('Amplitude');
title('Digital Data');
enc_data = [];
for i = 1:length(digi_data)
 if(digi_data(i)==1)
     m=m+1;
     if(rem(m,2) == 0)
        enc_data =[enc_data -1*ones(1,length(n))];
     else
         enc_data =[enc_data +1*ones(1,length(n))];
     end
 else
 enc_data =[enc_data zeros(1,length(n))];
 end
end
%----------------------------------------------------
figure(1),subplot(212),plot(t,enc_data,'LineWidth',2);
xlabel('time in sec');
ylabel('Amplitude');
title('Bipolar NRZ');
grid on;
```

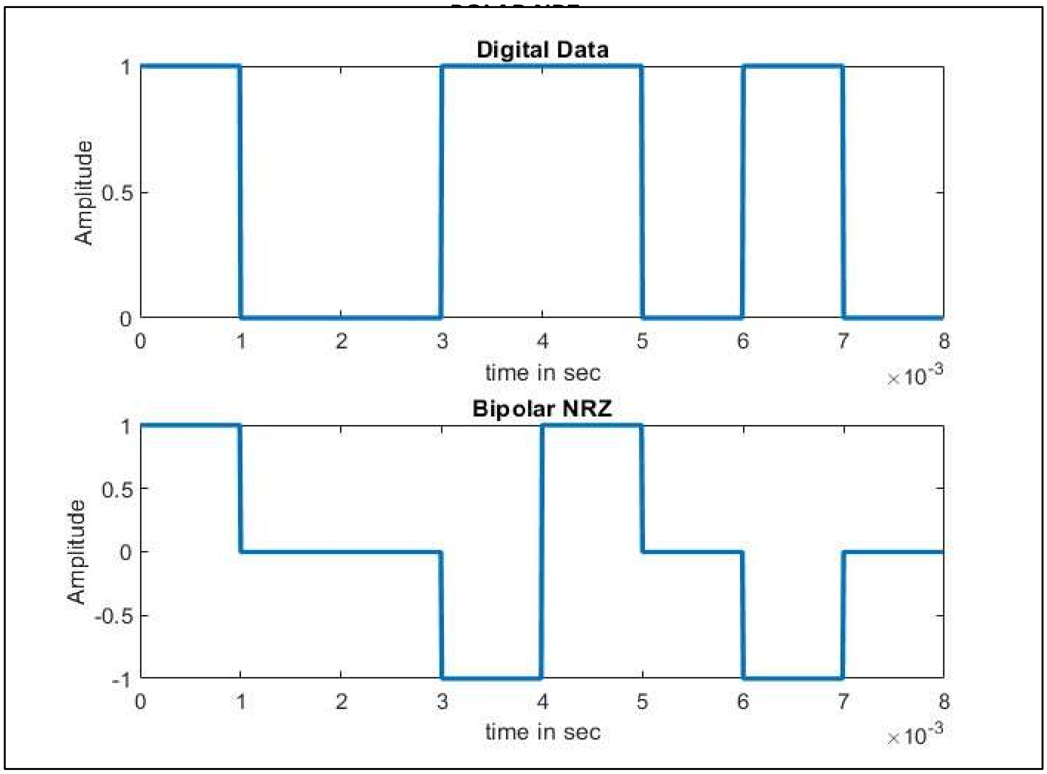
4. Manchester NRZ signal

```
clear all;close all;clc;

bit_d = 1e-3;% bit duration in sec
fc = 10000;% frequency of the carrier in Hz
fs = 10*fc;% Sampling rate
%-----------------------------------------------------------
n = 0:1:(bit_d*fs)-1;% Discrete time
m = 0:0.5:((bit_d*fs)-1);
carr1 = cos(2*pi*(fc/fs)*n);% Carrier wave for one bit duration
digi_data = [1 0 0 1 1 0 1 0];% digital data to be transmitted
carr = repmat(carr1,size(digi_data));% Carrier for entire
t = 0:1/fs:(length(carr)-1)/fs;% time for entire duration o
%----------------------
enc_digi = [];
for i = 1:length(digi_data)
 if(digi_data(i)==1)
 enc_digi =[enc_digi ones(1,length(n))];
 else
 enc_digi =[enc_digi zeros(1,length(n))];
 end
end
%------------------------------------------------------
figure(1),subplot(211),plot(t,enc_digi,'LineWidth',2);
xlabel('time in sec');
ylabel('Amplitude');
title('Digital Data');
enc_data = [];
for i = 1:length(digi_data)
 if(digi_data(i)==1)
 enc_data =[enc_data,1*ones(1,length(n)/2),-1*ones(1,length(n)/2)];
 else
 enc_data =[enc_data -1*ones(1,length(n)/2),1*ones(1,length(n)/2)];

 end
end

%------------------------------------------------------
figure(1),subplot(212),plot(t,enc_data,'LineWidth',2);
xlabel('time in sec');
ylabel('Amplitude');
title('Manchester NRZ');
```

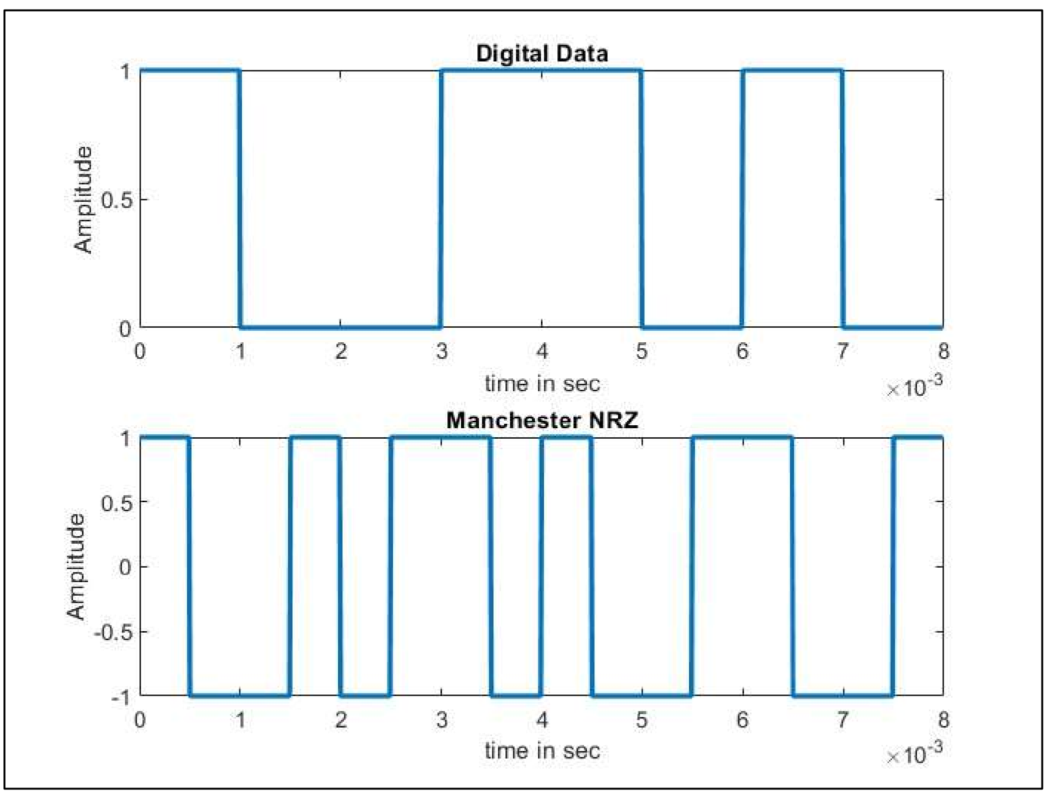


UNIPOLAR NRZ



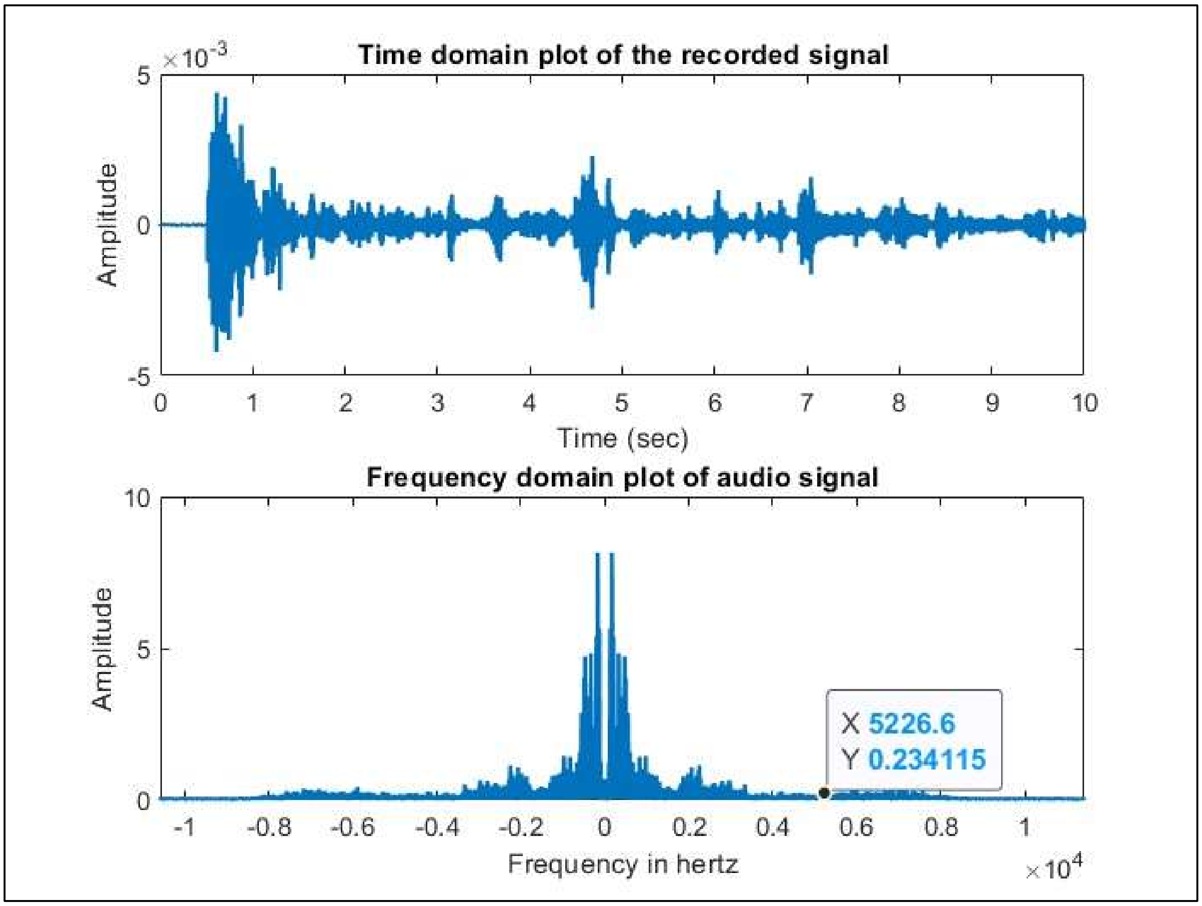

Digital Data
Amplitude
time in sec
×10^-3
Bipolar NRZ
Amplitude
time in sec
×10^-3

BIPOLAR NRZ


Digital Data
Amplitude
time in sec
×10^-3
Manchester NRZ
Amplitude
time in sec
×10^-3

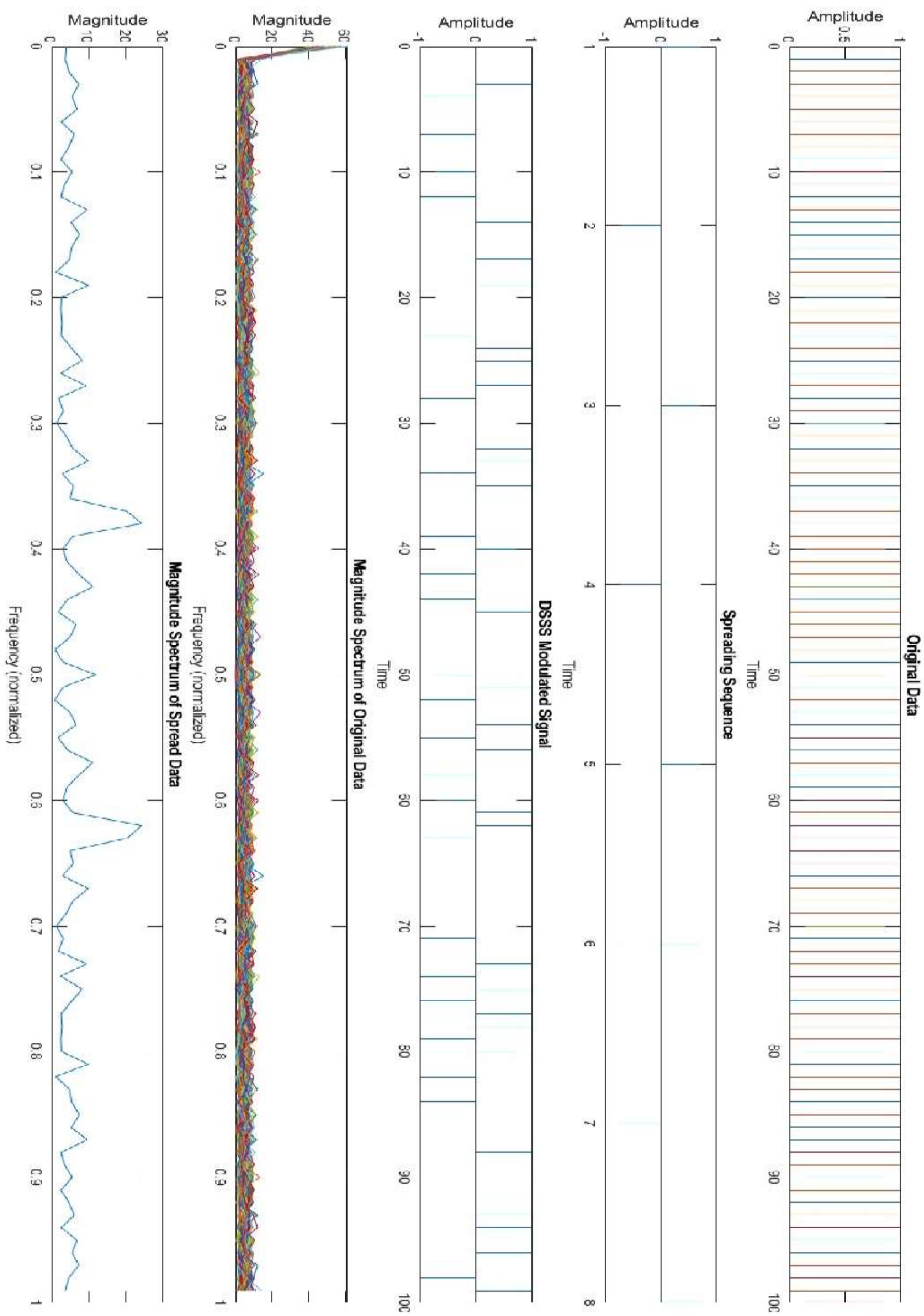
EXPT 1:

```
Fs=48000; Channels=1; bits=16;
r=audiorecorder(Fs,bits,Channels);
duration=10; disp('Recording started');
recordblocking(r,duration);
disp('Recording stopped');
x=getaudiodata(r);
%sound(x,Fs,bits);
t=0:1/Fs:(length(x)-1)/Fs;
subplot(2,1,1); plot(t,x,'LineWidth',1.5);
xlabel('Time (sec)'); ylabel('Amplitude');
title('Time domain plot of the recorded signal');
n=length(x); F=0:(n-1)*Fs/n;
y=fft(x,n);
F_0=(-n/2:n/2-1).*(Fs/n);
Y_0=fftshift(y);
AY_0=abs(Y_0);
subplot(2,1,2); plot(F_0,AY_0,'LineWidth',1.5);
xlabel('Frequency in hertz'); ylabel('Amplitude');
title('Frequency domain plot of audio signal');
filename='myvoice.wav';
audiowrite(filename,x,Fs);
```

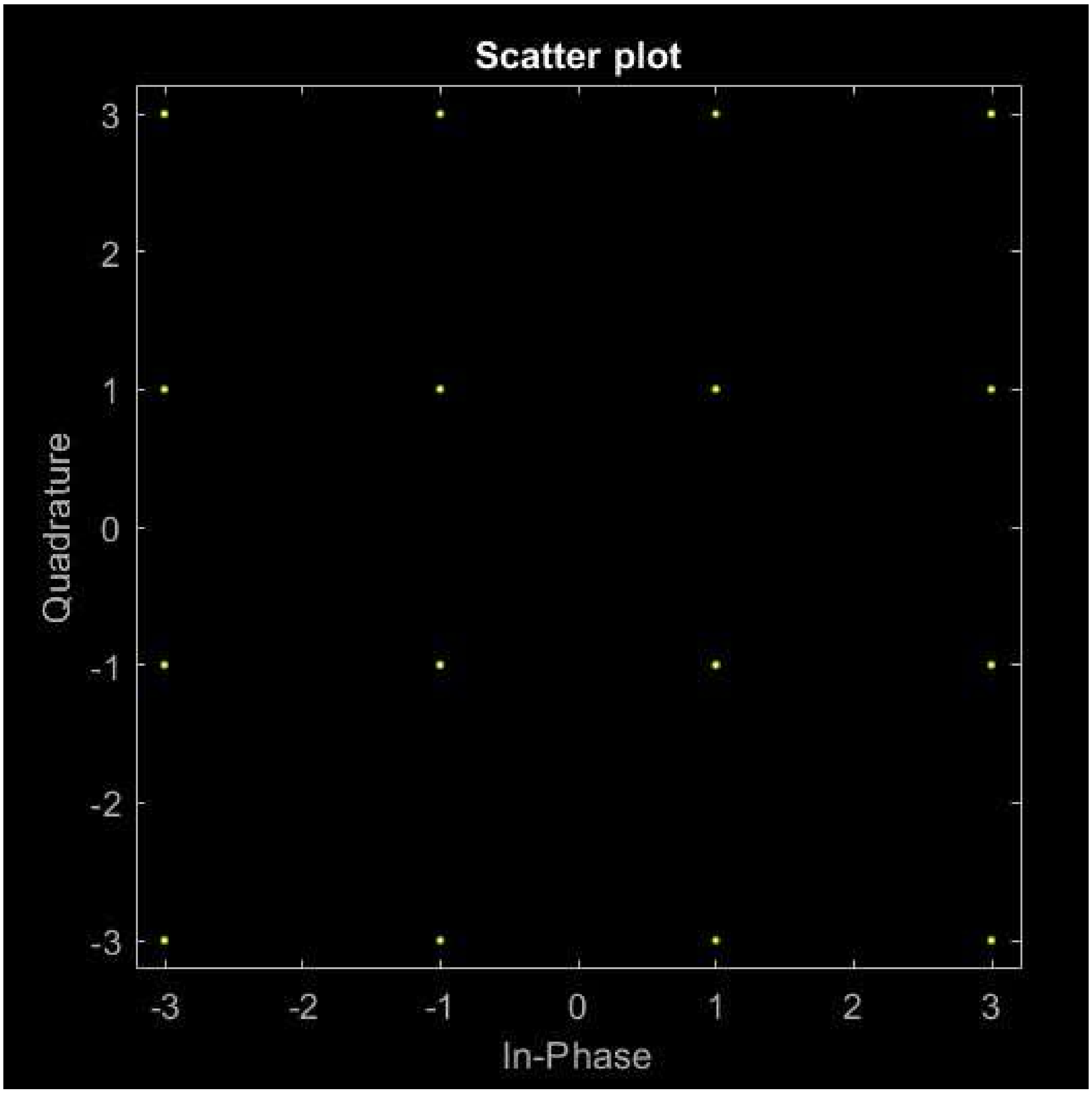

Time domain plot of the recorded signal
×10^-3
5
0
-5
Amplitude
0 1 2 3 4 5 6 7 8 9 10
Time (sec)
Frequency domain plot of audio signal
10
5
0
Amplitude
X 5226.6
Y 0.234115
-1 -0.8 -0.6 -0.4 -0.2 0 0.2 0.4 0.6 0.8 1
Frequency in hertz
×10^4

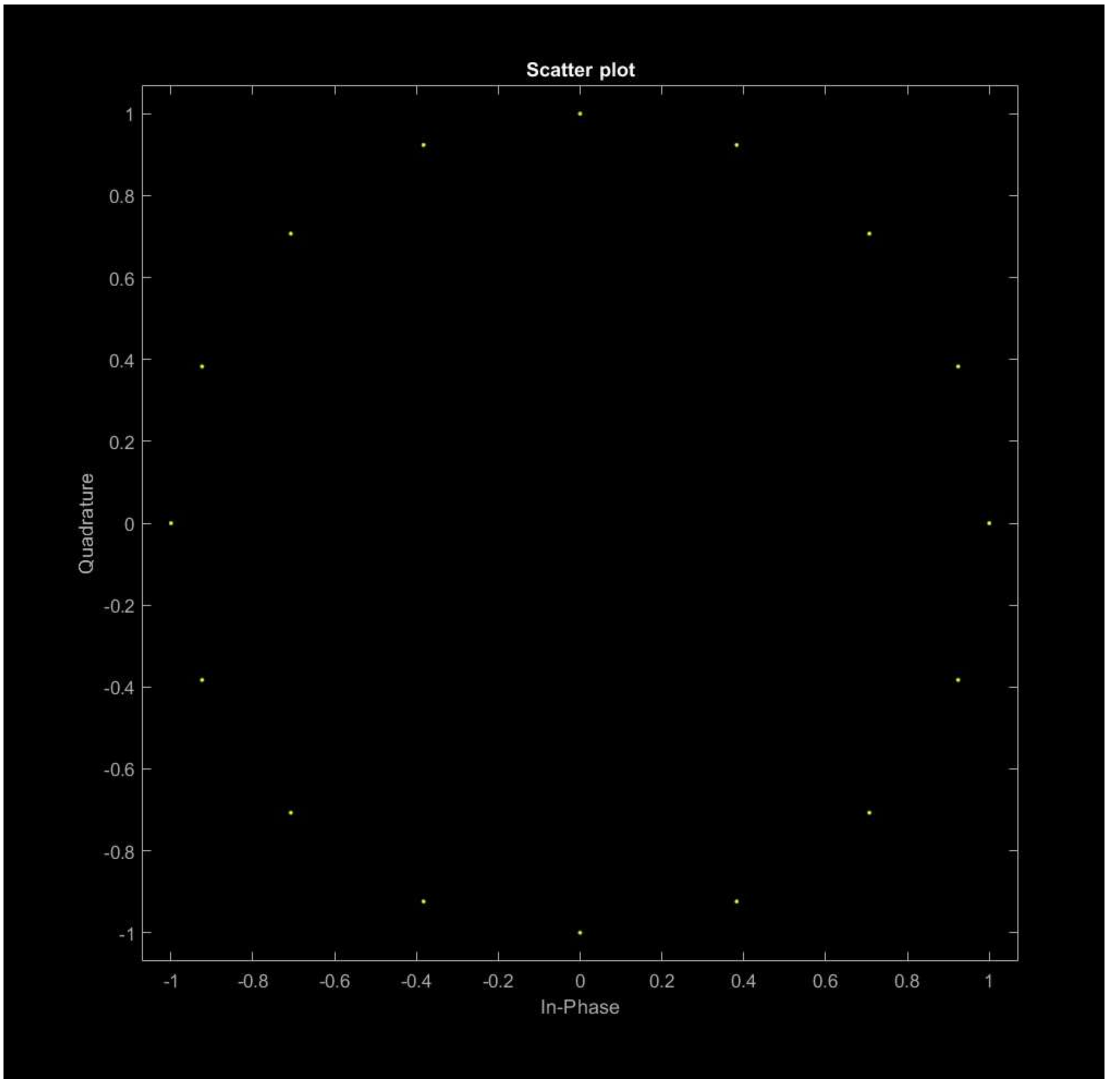
EXPT 10:

```
% Parameters
bitrate = 1000;           % Bitrate (bits per second)
T = 1 / bitrate;          % Sampling period
t = 0:T:1-T;              % Time vector
fc = 50;                  % Carrier frequency
A = 1;                    % Amplitude of data signal
chip_rate = 4 * bitrate;  % Chip rate
% Generate a random bit sequence
data = randi([0 1], 100);
% Create a spreading sequence
spreadingSequence = [1 -1 1 -1 1 1 -1 1];
% Spread the data using the spreading sequence
spreadData = zeros(1, length(data));
for i = 1:length(data)
    if i <= length(spreadingSequence)
        spreadData(i) = data(i) * spreadingSequence(i);
    else
        spreadData(i) = data(i) * spreadingSequence(mod(i-1,
length(spreadingSequence)) + 1);
    end
end
% Plot the original data, spreading sequence, DSSS modulated signal, and magnitude
spectrum
figure;
subplot(5,1,1);
stem(data, 'Marker', 'none');
xlabel('Time');
ylabel('Amplitude');
title('Original Data');
subplot(5,1,2);
stem(spreadingSequence, 'Marker', 'none');
xlabel('Time');
ylabel('Amplitude');
title('Spreading Sequence');
subplot(5,1,3);
stem(spreadData, 'Marker', 'none');
xlabel('Time');
ylabel('Amplitude');
title('DSSS Modulated Signal');
subplot(5,1,4);
magnitudeSpectrum = abs(fft(data));
f = (0:length(magnitudeSpectrum)-1)*(1/length(data));
plot(f, magnitudeSpectrum);
xlabel('Frequency (normalized)');
ylabel('Magnitude');
title('Magnitude Spectrum of Original Data');
subplot(5,1,5);
magnitudeSpectrum = abs(fft(spreadData));
f = (0:length(magnitudeSpectrum)-1)*(1/length(spreadData));
plot(f, magnitudeSpectrum);
xlabel('Frequency (normalized)');
ylabel('Magnitude'); title('Magnitude Spectrum of Spread Data');
```


Original Data
Amplitude
Time
Spreading Sequence
Amplitude
Time
DSSS Modulated Signal
Amplitude
Time
Magnitude Spectrum of Original Data
Magnitude
Frequency (normalized)
Magnitude Spectrum of Spread Data
Magnitude
Frequency (normalized)

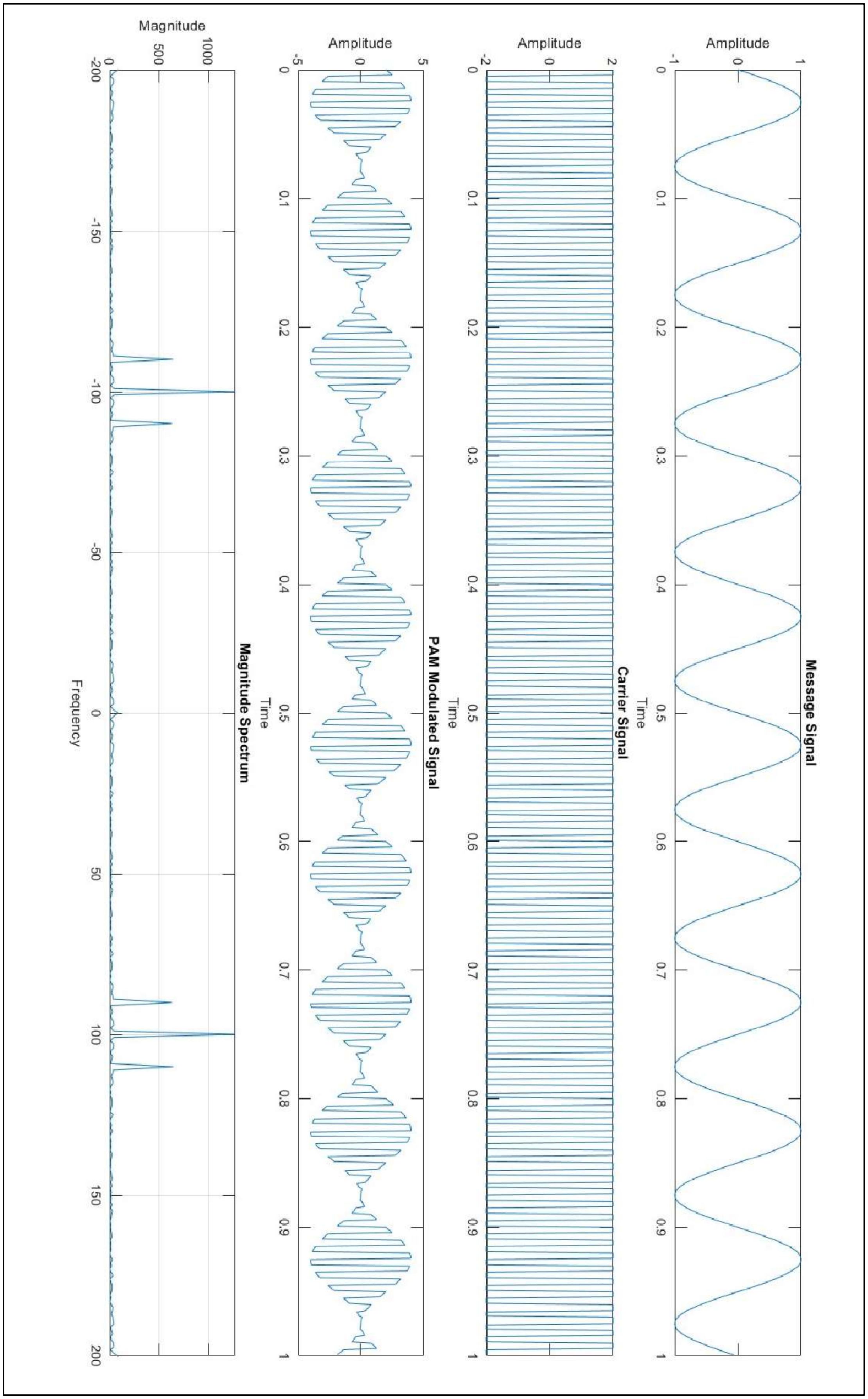
EXPT 8:




Scatter plot
Quadrature
In-Phase
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-1

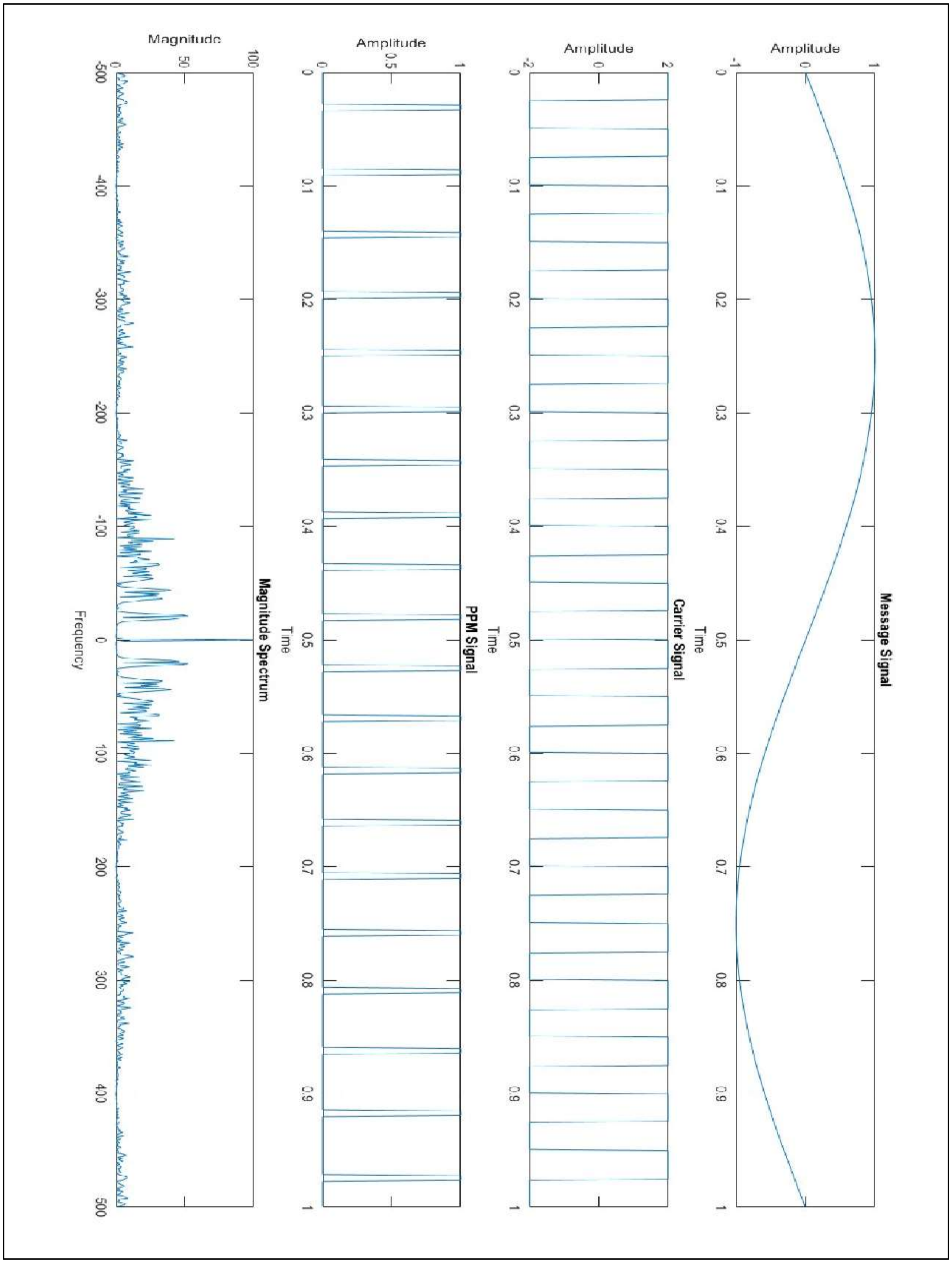
EXPT 2

```
Fs = 1000;          % Sampling frequency
T = 1/Fs;           % Sampling period
t = 0:T:1-T;        % Time vector

% Message signal
Am = 1;             % Amplitude of message signal
fm = 10;            % Frequency of message signal
message_signal = Am * sin(2*pi*fm*t);
subplot(4,1,1);
plot(t,message_signal);
title('Message Signal');
xlabel('Time');
ylabel('Amplitude');

% Carrier signal
Ac = 2;             % Amplitude of carrier signal
fc = 100;           % Frequency of carrier signal
carrier_signal = Ac * square(2*pi*fc*t);
subplot(4,1,2);
plot(t,carrier_signal);
title('Carrier Signal');
xlabel('Time');
ylabel('Amplitude');
% Modulation (PAM)
modulated_signal = (1 + message_signal) .* carrier_signal;

% Plotting the modulated signal
subplot(4, 1, 3);
plot(t, modulated_signal);
title('PAM Modulated Signal');
xlabel('Time'); ylabel('Amplitude');

% Frequency domain analysis
N = length(modulated_signal);
f = (-Fs/2):(Fs/N):(Fs/2-Fs/N);
magnitude_spectrum = abs(fftshift(fft(modulated_signal)));

% Plotting the magnitude spectrum
subplot(4, 1, 4);
plot(f, magnitude_spectrum);
title('Magnitude Spectrum');
xlabel('Frequency');ylabel('Magnitude');

% Adjust plot display
xlim([-200, 200]);  % Display frequency range -200 to 200 Hz
ylim([0, max(magnitude_spectrum)]);
grid on;
```


Message Signal
Amplitude
Time
Carrier Signal
Amplitude
Time
PAM Modulated Signal
Amplitude
Time
Magnitude Spectrum
Magnitude
Frequency

EXPT 2 b

```
% Carrier signal
Ac = 2;             % Amplitude of carrier signal
fc = 20;            % Frequency of carrier signal
duty=10;
Fs = 1000;          % Sampling frequency
T = 1/Fs;           % Sampling period
t = 0:T:1-T;
per=Fs/fc;
on_t = per/duty;
Am = 1;             % Amplitude of message signal
fm = 1;            % Frequency of message signal
message_signal = Am * sin(2*pi*fm*t);
subplot(4,1,1);
plot(t,message_signal);
title('Message Signal');
xlabel('Time');
ylabel('Amplitude');

carrier_signal = Ac * square(2*pi*fc*t);
subplot(4,1,2);
plot(t,carrier_signal);
title('Carrier Signal');
xlabel('Time');
ylabel('Amplitude');

A=1.25;
c=A.*sawtooth(2*pi*fc*t);%Carrier sawtooth
ppm = zeros(1,length(carrier_signal));
% find ids where carrier is greater than message
id = find(c > message_signal );
idd = diff(id);
iddd = find(idd ~= 1);
temp(1) = id(1);
temp(2:length(iddd)+1) = id(iddd + 1);
% ppm signal
for i = 1:length(temp)
    ppm(temp(i) : temp(i) +on_t - 1) = 1;
end
subplot(4,1,3);
plot(t,ppm);
title('PPM Signal');
xlabel('Time'); ylabel('Amplitude');
% Frequency domain analysis
N = length(ppm);
f = (-Fs/2):(Fs/N):(Fs/2-Fs/N);
magnitude_spectrum = abs(fftshift(fft(ppm)));

% Plotting the magnitude spectrum
subplot(4, 1, 4);
plot(f, magnitude_spectrum);
title('Magnitude Spectrum');
xlabel('Frequency'); ylabel('Magnitude');
```


Message Signal
Amplitude
Time
Carrier Signal
Amplitude
Time
PPM Signal
Amplitude
Time
Magnitude Spectrum
Magnitude
Frequency